Discrete Applied Mathematics 35 (1992) 29–45
North-Holland

Polynomial graph-colorings

Wolfgang Gutjahr

Institute für Informationsverarbeitung, Technische Universität Graz, Schießstattgasse 4a, A-8010 Graz, Austria

Emo Welzl and Gerhard Woeginger*

Institut für Informatik, Fachbereich Mathematik, Freie Universität Berlin, Arnimallee 2-6, D-1000 Berlin 33, Germany

Received 8 November 1988
Revised 1 September 1989

Abstract

Gutjahr, W., E. Welzl and G. Woeginger, Polynomial graph-colorings, Discrete Applied Mathematics 35 (1992) 29–45.

For directed graphs G and H, we say that G is H-colorable, if there is a graph homomorphism from G into H; that is, there is a mapping f from the vertex set of G into the vertex set of H such that whenever there is an edge (x, y) in G, then $(f(x), f(y))$ is an edge in H.

We introduce a new technique for proving that the H-coloring problem is polynomial time decidable for some fixed graphs H. Among others, this is the case if H is a semipath (a "path" with edges directed in either direction), which was not previously known. We also show the existence of a tree T, for which the T-coloring problem is NP-complete.

Keywords. Graph-coloring, NP-completeness, graph homomorphism.

1. Introduction

A *directed graph* G is a pair (V_G, E_G), where V_G is a set (the set of *vertices* of G) and $E_G \subseteq V_G \times V_G$ (the set of *edges* of G). For directed graphs G and H, an *H-coloring of G* is a mapping $f: V_G \to V_H$, such that for all edges $(x, y) \in E_G$ we have $(f(x), f(y)) \in E_H$. G is *H-colorable* if there exists an H-coloring of G.

If we carry the above definition of H-colorings to undirected graphs in the natural way, then K_n-coloring (K_n the complete graph on n vertices) coincides with n-coloring (i.e., coloring with n colors such that no two adjacent vertices get the same

* Current address: Wilhelm Gösser Gasse 13, A-8047 Graz, Austria.

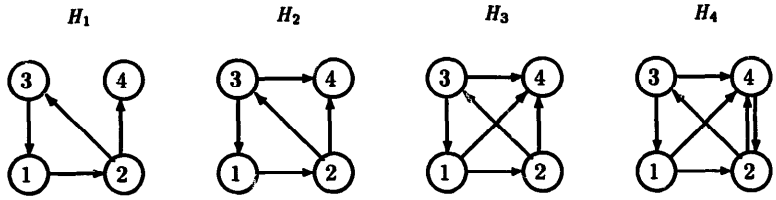

Fig. 1. A graph results from its predecessor by adding one edge. The first and third ones are polynomial graphs, the second and fourth ones are NP-complete.

color). From this analogy it follows that deciding whether G is H-colorable is NP-hard, for directed or undirected graphs G and H [2]. Obviously, the problem is in NP.

Here we are interested in the complexity of the H-coloring problem for fixed graphs H:

Instance: Graph G.
Question: Is G H-colorable?

Other names for this problem arising in the literature are "subgraph homomorphism problem" [2,5], "F-coloring problem" [8] or "graph interpretation problem" [6]. Hell and Nešetřil [4] have settled the undirected case, showing that H-coloring is NP-complete if H contains an odd cycle (and polynomial time decidable otherwise). However, for directed graphs, the situation is less clear. See Fig. 1 to advocate this impression: It gives a sequence of four graphs, where for $i=1,2,3$, H_{i+1} is obtained from H_i by adding exactly one edge. The first and third graph give rise to a polynomial coloring problem, the second and fourth graph are NP-complete. (We abuse notation in that we say a graph H is NP-complete if the H-coloring problem is NP-complete, and H is polynomial, if the corresponding coloring problem is.)

Convention: From now on, a graph is a directed graph!

The strongest result on the NP-complete side has been obtained by Bang-Jensen, Hell and MacGillivray [1]: A *semicomplete graph* (i.e., a graph without a loop and with at least one edge between any pair of vertices) is NP-complete if it contains at least two cycles and it is polynomial otherwise. Examples of polynomial graphs are [7]:

- *paths* $\vec{P}_n$, for $n \geq 0$: $\vec{P}_n$ is the graph with the vertex set $\{0,1,\ldots,n\}$ and the edge set $\{(i-1,i) \mid 1 \leq i \leq n\}$;
- *cycles* $\vec{C}_n$, for $n \geq 1$: $\vec{C}_n$ is the graph which is obtained from $\vec{P}_{n-1}$ by adding an edge from $n-1$ to 0;
- *transitive tournaments* $\vec{T}_n$, for $n \geq 1$: $\vec{T}_n$ is the transitive closure of $\vec{P}_{n-1}$;
- $H^{\leftarrow 1}$ *and* $H^{\rightarrow 1}$ *for polynomial graphs* H: $H^{\leftarrow 1}$ ($H^{\rightarrow 1}$) is obtained from H by adding a new vertex x, which has an edge to (respectively, from) all vertices in H.

The above graphs can be shown to be polynomial by more or less straightforward coloring algorithms. Note, e.g., that if $f(x)=i$ in a $\vec{P}_n$-coloring of a graph G, and (x, y) is an edge in G, then the color of y is completely determined by $f(y)=i+1$; similarly for $\vec{C}_n$. Actually, the above examples allow a "deterministic" build-up of a coloring.

Let us go through a few examples. A graph G is H_1-colorable (H_1 from Fig. 1) if and only if it is $\vec{C}_3$-colorable, and so H_1 is polynomial. Graph H_3 is isomorphic to $\vec{C}_3^{\rightarrow 1}$. A semicomplete graph with no cycle is a transitive tournament and a semicomplete graph with exactly one cycle can be obtained from $\vec{C}_2$ or $\vec{C}_3$ by repeated applications of the operations $^{\leftarrow 1}$ and $^{\rightarrow 1}$.

The goal of this paper is to extend the class of graphs which are polynomial. Perhaps most interesting, we show that *all semipaths are polynomial*, where a semipath is a graph obtained from a path by changing the direction of some of the edges. This is true even if the graph H (the semipath) is part of the input. The reader might argue that this is not too surprising in view of the simple structure of semipaths. Would one argue a similar case for trees? In order to put our result in a better perspective, we exhibit a tree T, for which the T-coloring problem is NP-complete.

Section 2 proposes a basic graph property, the so-called X-property, which ensures that the polynomial (actually linear) algorithm introduced in Section 3 works correctly. Section 4 extends the power of the algorithm to a larger class of graphs. Section 5 demonstrates that there is a tree T for which the T-coloring problem is NP-complete. Finally, a short discussion in Section 6 concludes the paper (among others, we cite here a graph that is polynomial but does not fit into our framework).

One more notion before we plunge into the rest of the paper. Graphs H_1 and H_2 are termed *color-equivalent* if, for all graphs G, G is H_1-colorable iff G is H_2-colorable. It is easily seen that this holds if and only if H_1 is H_2-colorable and H_2 is H_1-colorable. For example, if H_1 and H_2 contain a $\vec{C}_1$, then they are color-equivalent.

2. The graph property X

We start this section by introducing the most important definition of the paper.

Fig. 2. The X-graph H_5 together with its visualized X-enumeration.

Definition. Let H be a graph and let $(v_1, v_2, \ldots, v_n)$, $n = |V_H|$, be an enumeration of its vertices. We say, a pair (v_i, v_j) *dominates* a pair (v_k, v_l), or $(v_i, v_j) \geq (v_k, v_l)$, for short, iff $i \geq k$ and $j \geq l$ hold, and we say the pairs (v_i, v_j) and (v_k, v_l) are *crossing*, iff none of the two pairs dominates the other one (i.e. either $i > k$ and $j < l$, or $i < k$ and $j > l$). For pairs (v_i, v_j) and (v_k, v_l), the pair $(v_{\min(i,k)}, v_{\min(j,l)})$ is called the X-*pair* (spoken as: ex-underbar) of (v_i, v_j) and (v_k, v_l).

An enumeration of the vertices of H is called an X-*enumeration*, if for all pairs of edges (v_i, v_j) and (v_k, v_l) in E_H, the X-pair of (v_i, v_j) and (v_k, v_l) is in E_H, too. The graph H has the X-*property* (is an X-graph), if there exists an X-enumeration of its vertices.

In Fig. 2, an example for an X-graph is given. In order to visualize the X-property of a graph H, we consider two vertical bars with nodes 1 through $|V_H|$ bottom-up. An edge (v_i, v_j) is now represented by an arc from the node i on the left to node j on the right bar. Two edges (v_i, v_j) and (v_k, v_l) are crossing if and only if their corresponding arcs cross, and their X-pair is the "base" of the "X" determined by the crossing arcs. This explains the perhaps mysterious notation "X". In Fig. 2, the only crossing edges of H_5 are (2, 4) and (4, 3), and obviously, their X-pair (2, 3) is an edge, too, Hence, H_5 is an X-graph.

In the next section, we will show that deciding whether a graph G is H-colorable or not, can be done in polynomial time if H has the X-property. To simplify the presentation of our results, we introduce the following notation:

Notation. If H is an X-graph, then we assume that $V_H = \{1, 2, \ldots, n\}$, and that $(1, 2, \ldots, n)$ is an X-enumeration of V_H.

The following two lemmas show that all semipaths and all transitive tournaments belong to the family of X-graphs.

Lemma 2.1. *Every semipath P is an X-graph.*

Proof. An enumeration of a semipath obtained by starting at one vertex of degree one, walking through the semipath and ending at the other vertex of degree one is obviously an X-enumeration. □

Lemma 2.2. *For each $n \geq 2$, $\vec{T}_n$ has the X-property.*

Proof. For a transitive tournament $T = \vec{T}_n$, there exists an ordering $(t_1, t_2, \ldots, t_n)$ of V_T such that E_T can be defined as $(t_i, t_j) \in E_T$ iff $i < j$. We show that this ordering already is an X-enumeration. Let (i, j) and (k, l) be two crossing edges in E_T such that $i < k$ and $j > l$. By the ordering, $k < l$ must hold. This implies $i < k < l$, and thus the X-edge (i, l) is in E_T, too. □

Definition. For two colorings f_1 and f_2 of a graph G with colors $\{1,2,\ldots,n\}$, we write $f_1 \leq f_2$ if and only if $f_1(x) \leq f_2(x)$, for all vertices x of G. An H-coloring f is called the *minimum* H-coloring of G, if $f \leq f'$ holds, for all H-colorings f'.

Our algorithm will actually construct minimum H-colorings. To this end we will make use of the following two lemmas.

Lemma 2.3. *Let H be an $\underline{X}$-graph. For $x, y \in V_H$ (but (x,y) not necessarily an edge in E_H!), let $\mathcal{D}(x,y)$ denote the set of all edges in E_H that dominate (x,y). Then $\mathcal{D}(x,y)$ is either empty or contains a uniquely defined minimum element.*

Proof. If $\mathcal{D}(x,y)$ is empty, there is nothing to show. Hence, assume $\mathcal{D}(x,y)$ is not empty and contains two different minimal edges. But then the $\underline{X}$-pair of these edges, that is dominated by both of them, is in $\mathcal{D}(x,y)$, too; a contradiction. □

Lemma 2.4. *Let H be an $\underline{X}$-graph, $n = |V_H|$ and $(1,2,\ldots,n)$ be an $\underline{X}$-enumeration of its vertices. Then every H-colorable graph G has a minimum H-coloring f.*

Proof. We prove that for any two H-colorings f_1 and f_2 of G, there always exists an H-coloring f_3 of G, such that $f_3 \leq f_1$ and $f_3 \leq f_2$ holds. Similarly as in the proof of the last lemma, we then conclude that the set of all H-colorings of G cannot contain two different minimal colorings, nor is empty (as G is H-colorable).

We set $f_3(x) := \min\{f_1(x), f_2(x)\}$, for all vertices x in V_G. That means, every edge (x,y) that was colored by $(f_1(x), f_1(y))$ and $(f_2(x), f_2(y))$ (in f_1 and f_2, respectively), is now colored by the $\underline{X}$-pair of these two edges. By definition, this pair is in E_H, too. Obviously, $f_3 \leq f_1$ and $f_3 \leq f_2$ holds, and our proof is complete. □

3. The algorithm

We now present an algorithm, called $\underline{X}$-algorithm, for H-coloring a graph G which works correctly, if H has the $\underline{X}$-property.

We assume, as mentioned above, that $V_H = \{1,2,\ldots,n\}$. The algorithm will actually try to compute a minimum H-coloring f of G.

The $\underline{X}$-algorithm will use the following two basic "data structures": First, a coloring $\tilde{f}: V_G \to \{1,2,\ldots,n\}$ (which is not necessarily an H-coloring!) and secondly, a subset $\mathcal{C}$ of the edges in E_G. Throughout the algorithm, we will keep the following two invariants:

(i) If there is a minimum H-coloring f of G, then $\tilde{f} \leq f$.

(ii) For all edges (x,y) in $E_G - \mathcal{C}$, we have $(\tilde{f}(x), \tilde{f}(y)) \in E_H$.

In more intuitive terms, (i) means that $\tilde{f}$ can be turned into a minimum H-coloring by "increasing" the colors used (provided an H-coloring of G exists at all). Condition (ii) tells that the coloring $\tilde{f}$ is a valid H-coloring for the edges in $E_G - \mathcal{C}$, while the edges in $\mathcal{C}$ still have to be checked.

The initial structure is clear: We let $\tilde{f} \equiv 1$, and $\mathscr{C}$ is set equal E_G. Obviously, $\tilde{f}$ and $\mathscr{C}$ fulfill invariants (i) and (ii).

Then we start checking the edges in $\mathscr{C}$ in an arbitrary order. If the invariants are maintained all through the algorithm and $\mathscr{C}$ becomes empty some time, then because of condition (ii), it is clear that $\tilde{f}$ is a valid H-coloring of G. As long as $\mathscr{C} \neq \emptyset$, the checking of the edges in $\mathscr{C}$ is done in the following way:

Consider some edge (x, y) in $\mathscr{C}$. By Lemma 2.3, the set $\mathscr{D}(\tilde{f}(x), \tilde{f}(y))$ of all edges dominating $(\tilde{f}(x), \tilde{f}(y))$ in H is either empty or contains a uniquely defined minimum element. If $\mathscr{D}(\tilde{f}(x), \tilde{f}(y))$ is empty, the algorithm stops and says "G is not H-colorable". Otherwise, let (i, j) denote the minimum element in $\mathscr{D}(\tilde{f}(x), \tilde{f}(y))$.

Now, if $\tilde{f}(x) \neq i$ then we set $\tilde{f}(x) := i$ and we add all edges incident to x in G to the set $\mathscr{C}$. We proceed accordingly if $\tilde{f}(y) \neq j$. Then we remove (x, y) from $\mathscr{C}$ and go on checking.

To prove the correctness of the algorithm, all we have to do is to show that invariants (i) and (ii) are valid all through the algorithm and that the algorithm terminates in every case. The proof is done by induction on the number of checked edges.

(1) No edge checked until now: Then $\tilde{f} \equiv 1$ fulfills invariant (i) and $\mathscr{C} = E_G$ makes $E_G - \mathscr{C}$ fulfilling invariant (ii).

(2) Induction hypothesis. The invariants (i) and (ii) are still valid after the checking of $N \geq 0$ edges.

(3) Let (x, y) be the edge checked in Step $N+1$. For the unique minimum edge (i, j) in $\mathscr{D}(\tilde{f}(x), \tilde{f}(y))$, we observe that $f(x) \geq i$ and $f(y) \geq j$ holds for the minimum H-coloring f of G, provided it exists. Thus condition (i) is maintained in Step $N+1$. All edges that are incident to vertices with a new color after Step $N+1$ are added to $\mathscr{C}$ (except for (x, y)). From the induction hypothesis and the fact that (x, y) is correctly colored, we thus get that condition (ii) still holds after Step $N+1$. This completes the inductive proof.

If some edge (x, y) is added to $\mathscr{C}$, the color of at least one of its incident vertices x, y is increased. Hence, each edge is added to $\mathscr{C}$ at most $2n-2$ times and checked at most $2n-1$ times. If H is preprocessed, the minimum edge in $\mathscr{D}(\tilde{f}(x), \tilde{f}(y))$ can be found in constant time. If we consider an edge for addition in $\mathscr{C}$ (even though it might be already in $\mathscr{C}$), then one of the colors of the incident vertices has been increased. Hence, the whole algorithm terminates after at most $(2n-1) \cdot |E_G|$ checking steps in $O(n|E_G| + |V_G|)$ time.

We summarize the result in the following theorem:

Theorem 3.1. *For each X-graph H with n vertices, there exists an algorithm that decides for any graph G, whether it is H-colorable or not and constructs the H-coloring, if it exists, in $O(n|E_G| + |V_G|)$ time.*

Due to Lemma 2.1, the theorem implies that the P-coloring problem is polynomial for all semipaths P. Actually, a stronger statement can be made.

Theorem 3.2. *There is an algorithm that decides for any graph G and for any semipath P, whether G is P-colorable and constructs a P-coloring, if it exists, in $O(|V_P| \cdot |E_G| + |V_G|)$ time.*

Proof. Clearly, the $\underline{X}$-enumeration of a semipath P indicated in the proof of Lemma 2.1 can be found in $O(|V_P|)$ time. For a pair (k,l), the minimum edge in $\mathcal{D}(k,l)$ is one of the edges incident to $\max\{k,l\}$ (the verification of this assertion is left to the reader). Now the theorem follows along the lines of the above algorithm. □

We finish this section with a lemma, that restricts the family of $\underline{X}$-graphs basically to a subset of the acyclic graphs.

Lemma 3.3. *Let H be an $\underline{X}$-graph. If H contains any cycle, then it is color-equivalent to $\vec{C}_1$.*

Proof. We show that if H contains a cycle then it must contain a $\vec{C}_1$. To this end, let $(1,2,\ldots,n)$ be an $\underline{X}$-enumeration of E_H. Let i be the smallest vertex in this enumeration lying in a cycle, let j_1 and j_2 be its neighbors in the cycle. Then E_H contains the edges (i,j_1) and (j_2,i) with $i \leq j_2$ and $j_1 \geq i$ and thus, the $\underline{X}$-pair (i,i) must be in E, too.

Hence, H contains a loop and any graph G admits an H-coloring - map all vertices of G to the loop-node. Thus it is color-equivalent to $\vec{C}_1$. □

4. Extending the algorithm

As it was seen, the family of $\underline{X}$-graphs already contains quite a number of graphs which thus are polynomial. But it does not contain the cycles and other graphs that are known to be polynomial. For example, the graph $H_6=(V_6,E_6)$, where $V_6=\{v_1,v_2,v_3,v_4\}$ and $E_6=\{(v_1,v_2),(v_2,v_3),(v_3,v_4),(v_1,v_4)\}$, is easily checked not to be an $\underline{X}$-graph (see Fig. 3). We will extend our technique to show that H_6 is also polynomial.

4.1. The cycle extension

Consider the enumeration of V_6 in Fig. 3(a) which is not an $\underline{X}$-enumeration, but nevertheless has other nice properties.

The first crucial observation is that H_6 is bipartite (i.e., $\vec{C}_2$-colorable), as there are only edges between the sets $\{1,3\}$ and $\{2,4\}$.

The other important fact is that the edges in H_6 can be organized as in Fig. 3(b): The set of edges going from $\{1,3\}$ to $\{2,4\}$ has the $\underline{X}$-property and the set of edges going from $\{2,4\}$ to $\{1,3\}$ has it, too. This makes H_6 a special instance of the graph family described in the following definition.

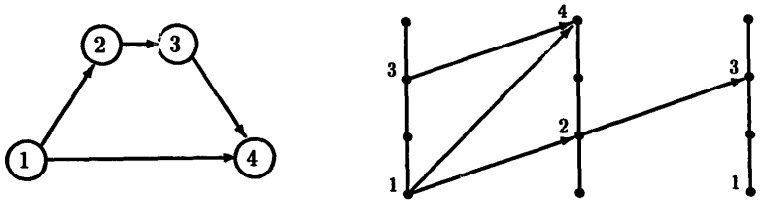

Fig. 3. (a) The graph H_6; and (b) an representation of its edges.

Definition. A graph $H=(V,E)$ is said to have the $\vec{C}_k$-*extended* X-*property*, $k\geq 1$, if it fulfills the following two conditions:

(i) H is $\vec{C}_k$-colorable. Now let $V_H^1, V_H^2, \ldots, V_H^k$ denote the partition of V induced by an $\vec{C}_k$-coloring, such that

$$E \subseteq \bigcup_{i=1}^{k-1} (V_H^i \times V_H^{i+1}) \cup (V_H^k \times V_H^1).$$

(ii) There exists an enumeration of the vertices in V such that each subgraph of H with edge set $E \cap (V_H^i \times V_H^{i+1})$, $1 \leq i \leq k-1$, or $E \cap (V_H^k \times V_H^1)$ has the X-property (together with its induced enumeration).

In connection with item (i) of the above definition, it should be mentioned that the used partition is unique up to a circular rotation of the V_H^i for weakly connected graphs. This can be seen easily: If we have chosen the color of one vertex, this vertex uniquely determines the colors of all the other vertices. At the same time, this simple observation gives rise to an algorithm, testing for a $\vec{C}_k$-coloring and constructing it, that runs in polynomial time.

Note that the $\vec{C}_1$-extended X-property is equivalent to the X-property.

For $m, n \geq 1$, let $\vec{P}_{n,m}$ denote the graph that results from glueing together the paths $\vec{P}_n$ and $\vec{P}_m$ at their starting and end vertices. Obviously, the graph $\vec{P}_{n,n}$ is color-equivalent to $\vec{P}_n$ and therefore a polynomial graph. All other $\vec{P}_{n,m}$ are $\vec{C}_k$-extended X-graphs, for some k (and thus polynomial graphs, as it will be shown in this section).

Lemma 4.1. *Let $n, m \geq 1$, $n \neq m$. Then $\vec{P}_{n,m}$ is a $\vec{C}_k$-extended X-graph, for some integer $k \geq 1$.*

Proof. W.l.o.g. we assume $n > m$. Let $n - m = p$. As $n+1 (\mathrm{mod}\, p) = m+1 (\mathrm{mod}\, p)$ holds, $\vec{P}_{n,m}$ is $\vec{C}_p$-colorable. We define an enumeration of $\vec{P}_{n,m}$ in the following way.

The vertices on $\vec{P}_n$ get the numbers $\{1, 2, 3, 4, \ldots, p, p+1, p+3, \ldots, p+1+2(m-1) = n+m-1, n+m\}$ and the vertices on $\vec{P}_m$ get the numbers $\{1, p+2, p+4, \ldots, p+2(m-1) = n+m-2, n+m\}$. It is easily checked that this enumeration is a $\vec{C}_p$-extended X-enumeration. □

Theorem 4.2. *For every graph H with n vertices that has the $\vec{C}_k$-extended $\underline{X}$-property, there exists an algorithm that tests for any graph G, whether G is H-colorable or not and constructs an H-coloring (if it exists) in $O(k \cdot n \cdot |E_G| + |V_G|)$ time.*

Proof. We prove the theorem by explicitly giving the algorithm. We use the notation introduced in the above definition. Let $(1,2,\ldots,n)$ be an $\vec{C}_k$-extended $\underline{X}$-enumeration of V_H. For $x \in V_H$, we denote by $\text{set}(x)$ that one of the sets $V_H^1, V_H^2, \ldots, V_H^k$, for which $x \in \text{set}(x)$ holds.

As H is $\vec{C}_k$-colorable, every graph G colorable by H must be $\vec{C}_k$-colorable, too. Thus we start by testing, whether G is $\vec{C}_k$-colorable. If it is not, we stop with the result "G is not H-colorable", if it is, we calculate the partition $V_G^1, V_G^2, \ldots, V_G^k$ of V_G induced by this coloring.

We now consider some fixed vertex x_1 in V_G^1. x_1 must be colored by some color in one of the sets V_H^i, $1 \le i \le k$, and we will only examine the case where it gets a color in V_H^1. (If we do not succeed in finding an H-coloring in this case, we check the cases where it gets a color in $V_H^2, V_H^3, \ldots$ and so on. If we discover a valid coloring, everything is alright, if we do not, no coloring can exist.) Our assumption implies that all vertices in V_G^i must get a color in V_H^i, $1 \le i \le k$.

We are ready to give the adapted $\underline{X}$-algorithm.

- The initial structure is changed as follows: For $x \in V_G^i$, $\tilde{f}(x)$ is initially set to the smallest color in V_H^i. $\mathscr{C}$ is set to E_G again.
- To modify the edge-checking step, we only need to change the set of dominating edges $\mathscr{D}(x,y)$ to $\mathscr{D}(x,y) \cap (\text{set}(x) \times \text{set}(y))$. All the rest is done exactly the same way as in the $\underline{X}$-algorithm.

Why does our modified algorithm work correctly?

The only thing to say is that we restrict the colors possible for a vertex to those that actually can be used to color it. If x is in V_G^i, it only can be colored by a color in V_H^i. Hence, it does not make sense to initialize $\tilde{f}(x)$ with any color outside of V_H^i, and thus it is set to the smallest possible color in this set. Analogously, if $x \in V_G^i$ and $y \in V_G^{i+1}$ holds, only dominating edges need to be considered that appear between V_H^i and V_H^{i+1}.

Constructing a $\vec{C}_k$-coloring for G (if one exists) can be done in time proportional to $|E_G|$. As, in the worst case, the whole extended algorithm consists of such a construction together with k applications of the $\underline{X}$-algorithm, its running time lies in the claimed $O(k \cdot n \cdot |E_G|)$ bound. □

Obviously, $\vec{C}_k$-extended $\underline{X}$-graphs *can* contain cycles and we have eliminated that weak point of the $\underline{X}$-graphs. But the next lemma gives us a throwback, again.

Lemma 4.3. *Let H be a $\vec{C}_k$-extended $\underline{X}$-graph. If H contains any cycle, then it is color-equivalent to $\vec{C}_k$.*

Proof. By definition, H is $\vec{C}_k$-colorable and thus the length of any cycle in H must be a multiple of k. Now let $V_H^1, V_H^2, \ldots, V_H^k$ denote the partition of V induced by this $\vec{C}_k$-coloring. Consider an $\vec{C}_k$-extended $\underline{X}$-enumeration of H. Let $\vec{C}$ be any cycle in H and for $1 \leq i \leq k$, let v_i be the smallest number in the enumeration that is contained in $V_H^i \cap V_{\vec{C}}$. Because of the $\underline{X}$-property that holds between the classes V_H^i and V_H^{i+1}, the sequence $(v_1, v_2, \ldots, v_k)$ itself forms a cycle in H.

Hence, the $\vec{C}_k$-colorable graph H contains a $\vec{C}_k$ as subgraph and so H is color-equivalent to $\vec{C}_k$. □

We finish this section with the observation that the $\underline{X}$-property is a special case of the $\vec{C}_k$-extended $\underline{X}$-property (if k is set equal one).

4.2. The graft extension

Although we extended the $\underline{X}$-property, there is still a very simple polynomial graph not covered yet. The graph $H_7 = (V_7, E_7)$, where $V_7 = \{1,2,3,4\}$ and $E_7 = \{(1,2),(2,3),(2,4),(3,4),(4,3)\}$ (see Fig. 4), can be shown to be polynomial, and still it does not belong to the extended $\underline{X}$-class.

If we contract the vertices 3 and 4 in H_7 to a single vertex, we receive the directed path $\vec{P}_2$. This similarity to $\vec{P}_2$ will be used to drag H_7 on our side. We start with a definition.

Definition. Let $J = (V_J, E_J)$ be a loopfree $\underline{X}$-graph with $\underline{X}$-enumeration $(1, 2, \ldots, n)$. Let $K = (V_K, E_K)$ be a polynomial digraph. W.l.o.g. let $V_J \cap V_K = \emptyset$.

$\underline{X}$-graft(J, K) is the graph $H = (V_H, E_H)$ defined as follows:

$$V_H = V_J \cup V_K - \{n\}$$

(i.e., the union of both vertex sets without the vertex with highest $\underline{X}$-number in J)

$$\begin{aligned} E_H = E_K &\cup \{(x,y) \mid (x,y) \in E_J,\ x \neq n,\ y \neq n\} \\ &\cup \{(x,y) \mid x \in V_J,\ y \in V_K,\ (x,n) \in E_J\} \\ &\cup \{(x,y) \mid x \in V_K,\ y \in V_J,\ (n,y) \in E_J\} \end{aligned}$$

(i.e., the edges in K and the edges in J that are not incident to n, together with all edges from a vertex x to K, if there is an edge in J going from x to n, together with all edges from K to a vertex x, if there is an edge in J from n to x).

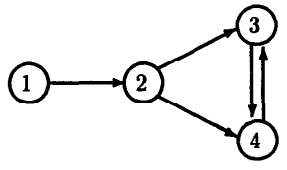

Fig. 4. The polynomial graph H_7.

If we contract the subgraph K in $\underline{X}$-graft(J,K) to a single vertex, the resulting graph is isomorphic to J. Obviously, H_7 is the graph $\underline{X}$-graft$(\vec{P}_2, \vec{C}_2)$.

Note that the definition of $\underline{X}$-graft(J,K) depends on the $\underline{X}$-enumeration of J, which is not unique, in general. If we consider graphs $\underline{X}$-graft$(\vec{P}_1, H)$, then we recognize a familiar operation from the introduction:

Lemma 4.4. *Let H be a graph and $\vec{P}_1 = (\{v_1, v_2\}, \{(v_1, v_2)\})$. Then $\underline{X}$-graft$(\vec{P}_1, H)$ with $\underline{X}$-enumeration (v_1, v_2) equals $H^{\leftarrow 1}$ and with $\underline{X}$-enumeration (v_2, v_1) equals $H^{\rightarrow 1}$.*

Proof. Obvious. □

Theorem 4.5. *If J is an $\underline{X}$-graph without loops and K is a polynomial graph, then coloring by $H = \underline{X}$-graft(J,K) is polynomial time decidable.*

Proof. We analyze, whether some fixed graph $G = (V_G, E_G)$ is H-colorable or not.

First, we apply the $\underline{X}$-algorithm to G: We try to $\hat{J}$-color it by the graph $\hat{J}$ that results from J by adding the edge (n,n). As (n,n) is the "highest" edge possible in $E_{\hat{J}}$, no other edge can cross it and thus $\hat{J}$ is an $\underline{X}$-graph again. The only problem appearing here is, that $\hat{J}$ contains a loop. Hence, *all* graphs are $\hat{J}$-colorable, as we can simply give color n to each vertex. The crucial fact is that we apply the $\underline{X}$-algorithm and the $\underline{X}$-algorithm only constructs *minimum* colorings. What does this mean? All vertices in G that are colored by n now, can never get a color lower than n in a valid H-coloring, and thus they must get some color in K.

Hence, we try to K-color the subgraph $G(n)$ of G that is induced by the n-colored vertices. This can be done in polynomial time, as K is a polynomial graph, by assumption. If $G(n)$ is not K-colorable, we stop and say "G is not $\underline{X}$-graft(J,K)-colorable". If $G(n)$ *is* K-colorable, then G is $\underline{X}$-graft(J,K)-colorable, as the following demonstrates. Let $\tilde{f}$ denote the coloring at which we are arrived now and let (x,y) be some edge in G:

(a) If $\tilde{f}(x)$ and $\tilde{f}(y)$ are both in V_J, then $(\tilde{f}(x), \tilde{f}(y))$ is in $E_J \subseteq E_H$, because these colors were only allocated in the $\underline{X}$-algorithm.

(b) If $\tilde{f}(x)$ and $\tilde{f}(y)$ are both in V_K, then $(\tilde{f}(x), \tilde{f}(y))$ is in $E_K \subseteq E_H$, because these colors were only used when G was colored by K.

(c) If $\tilde{f}(x)$ is in V_J and $\tilde{f}(y)$ is in V_K, then the edge $(\tilde{f}(x), n)$ is in E_J, as the color n was given to y by the $\underline{X}$-algorithm. Hence, $(\tilde{f}(x), \tilde{f}(y))$ is in E_H.

(d) If $\tilde{f}(x)$ is in V_K and $\tilde{f}(y)$ is in V_J, we have a case symmetric to (c). x got color n in the $\underline{X}$-algorithm and thus, $(\tilde{f}(x), \tilde{f}(y))$ is in E_H.

Summarizing, we can say that G is $\underline{X}$-graft(J,K)-colorable if and only if the induced subgraph $G(n)$ is K-colorable. □

It can be easily shown that every semicomplete graph with exactly one cycle can be obtained from $\vec{C}_2$ or $\vec{C}_3$ by repeated applications of the operations $^{\rightarrow 1}$ and $^{\leftarrow 1}$.

By Lemma 4.4, this shows that these graphs fit into our framework.

Though an $\underline{X}$-graft(J,K) can contain a cycle and need not be isomorphic to a cycle, it cannot produce any new cycles outside of K.

Lemma 4.6. *Let $H=(V_H,E_H)$ be an $\underline{X}$-graft(J,K). Then H contains the same number of cycles as K.*

Proof. Assume, H would contain more cycles than K. Then parts of some cycle $\vec{C}$ in H must lie outside of K. If we contract the subgraph K of H, the cycle parts outside of K produce a cycle in the resulting graph. But the resulting graph is the $\underline{X}$-graph J that does not contain any loop and, hence, by Lemma 3.3, it cannot contain any cycle; a contradiction. □

5. An NP-complete tree

All we do in this section is to construct a directed tree T such that the T-coloring problem is NP-complete. Unfortunately, the tree T has 287 vertices; hence, some preparation work is in place.

First, consider a weakly connected graph G which is $\vec{P}_n$-colorable but not $\vec{P}_{n-1}$-colorable. Then the $\vec{P}_n$-coloring of G is unique. Recall that $V_{\vec{P}_n}=\{0,1,\ldots,n\}$ and $E_{\vec{P}_n}=\{(i-1,i)\mid 1\le i\le n\}$. The *index* of a vertex x in G, denoted $i_G(x)$, is the color it gets in a $\vec{P}_n$-coloring.

Lemma 5.1. *Let G and H be weakly connected graphs which are $\vec{P}_n$-colorable but not $\vec{P}_{n-1}$-colorable. If f is an H-coloring of G, then $i_H(f(x))=i_G(x)$ for all vertices x in G.*

Proof. Note that i_G and i_H are $\vec{P}_n$-colorings of G and H, respectively, which are unique, as we observed before. Since the composition of i_H and f is a $\vec{P}_n$-coloring of G, too, the assertion follows. □

Our tree will consist of "superedges" and "supervertices"; the superedges are semipaths or trees composed of "normal" edges and vertices. To this end let P_i, $1\le i\le 6$, be the semipath consisting of $i+1$ edges forwards, then one edge backwards and then $9-i$ edges forwards, again. The vertex of degree one with an outgoing edge is called *source* of P_i and the other vertex of degree one is called *target* of P_i. Our *superedges* are now defined as follows (see Fig. 5):

- $\alpha=P_1$, $\beta=P_2$;
- $a=P_3$, $b=P_4$, $c=P_5$, $e=P_6$;
- e_1 is obtained from e by adding a vertex and an edge pointing from the unique vertex in e with two ingoing edges to the new vertex; and e_2 is obtained from

e by adding a vertex with an edge pointing towards the unique vertex in *e* with two outgoing edges;

- the definition of *source* and *target* of those superedges is carried over from their underlying semipath P_i.

We use *s-edge* short for superedge, and we call the source and target of an s-edge the *s-vertices* incident to this edge.

All s-edges defined are $\vec{P}_9$-colorable but not $\vec{P}_8$-colorable; the index of each source is 0 and of each target is 9. Moreover, no vertex inside an s-edge has index 0 or 9 (see Fig. 5). Note that *e* is subgraph of e_1 and of e_2, and so *e* is both, e_1- and e_2-colorable. No other colorings are possible among s-edges, which can be seen as follows: If s-edge σ is τ-colorable for some s-edge τ, then source (and target) of σ is colored by source (and target) of τ (by Lemma 5.1). The (semi)path between source and target has 11 edges in all s-edges; consequently, all colors in an τ-coloring

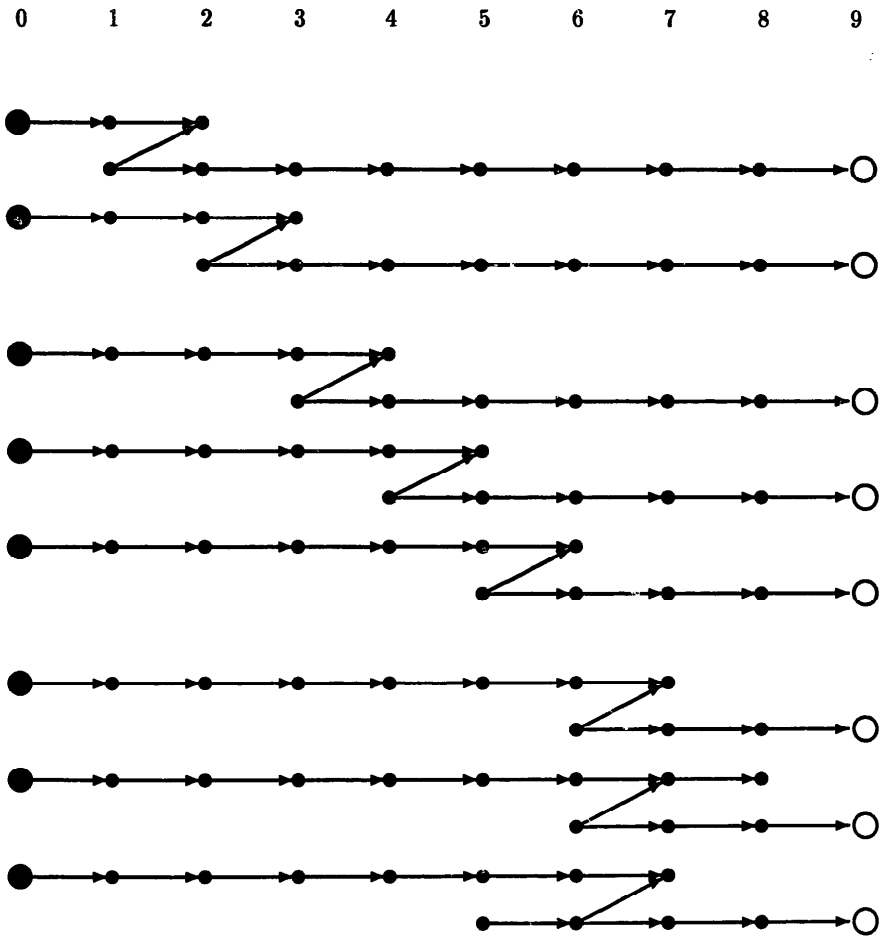

Fig. 5. The superedges α, β, a, b, c, e, e_1, and e_2.

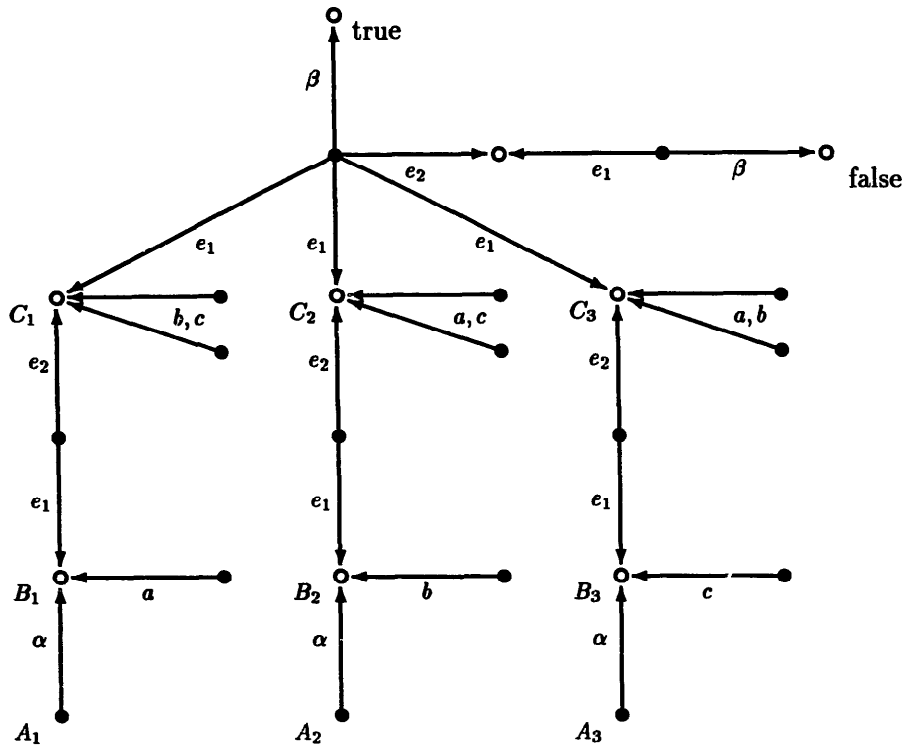

Fig. 6. The NP-complete tree T. Black vertices are s-vertices with index 0, white (empty) vertices are s-vertices with index 9.

of σ are completely determined for all vertices on this semipath, and they are legal only if these semipaths are the same in σ and τ. If $\sigma \neq \tau$, this is only possible among s-edges e, e_1, and e_2. Now it is easily seen that the additional vertices in e_1 and e_2 prohibit all colorings except for e_1- and e_2-colorings of e.

The tree T is given in Fig. 6: Only s-vertices are explicitly displayed and the s-edges are represented by arcs (from source to target) with labels indicating their types. T is $\vec{P}_9$-colorable; black s-vertices are vertices in T with index 0 and white s-vertices are vertices in T with index 9. The only vertices in T with index 0 or 9 are the s-vertices.

The properties we have developed above for the s-edges entail the following properties for T:

Lemma 5.2. *Let σ be an s-edge and let f be a T-coloring of σ. For the source x of σ, $f(x)$ is a black s-vertex in T and for the target y of σ, $f(y)$ is a white s-vertex in T.*

If $\sigma = e$, then $f(x)$ and $f(y)$ are connected by an s-edge of type e_1 or e_2 in T. If $\sigma \neq e$, then $f(x)$ and $f(y)$ are connected by an s-edge of type σ in T.

We demonstrate that the ONE-IN-THREE 3SAT problem is polynomial time

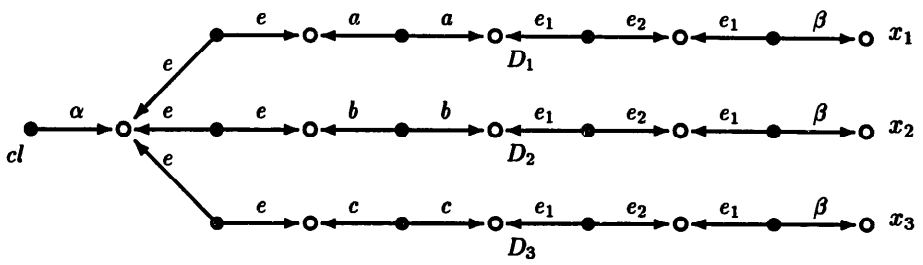

Fig. 7. The graph that corresponds to the clause $\mathrm{cl} = (x_1, x_2, x_3)$.

reducible to the T-coloring problem. As ONE-IN-THREE 3SAT is known to be NP-complete [2], this will imply that T-coloring is NP-complete, too.

ONE-IN-THREE 3SAT

Instance: A set U of variables and a collection C of clauses over U, such that each clause $\mathrm{cl} \in C$ consists of exactly three, nonnegated literals.

Question: Is there a truth assignment for U such that each clause in C has exactly one true literal?

For a ONE-IN-THREE 3SAT problem $P = (U, C)$, we give a graph G that is T-colorable iff P has a solution. To this end, G contains for each literal $x \in U$ and for each clause $\mathrm{cl} \in C$ a vertex x or cl, respectively. Let $\mathrm{cl} = (x_1 + x_2 + x_3)$ be some clause in C. We connect the vertices cl, x_1, x_2, x_3 in the graph G as demonstrated in Fig. 7 (we take the clauses to be ordered triples). Hence, G is $\vec{P}_9$-colorable. All vertices corresponding to clauses have index 0, all vertices corresponding to literals have index 9.

Which colors can a vertex cl get in a T-coloring? As an s-edge α points away from cl, cl must be colored by A_1, A_2 or A_3. Analogously, each vertex x must get color **true** or **false**, as an edge β points to it.

Now we take a closer look at the case "cl is colored by A_1":

- As there is an s-edge a pointing to the vertex D_1, D_1 must get one of the colors B_1, C_2 or C_3. Moreover, colors C_2 and C_3 can immediately be ruled out, as they are too far away from vertex A_1. This implies that D_1 is colored by B_1. As there are only four s-edges between vertex D_1 and x_1 (and all s-edges have the same length), color **false** cannot be reached anymore. Hence, x_1 must be colored by **true**.
- Similarly as we got B_1 to be the only color possible for vertex D_1, we can show that D_2 must be colored by C_1. Now the coloring of the s-vertices between D_2 and x_2 is completely determined, and x_2 must be colored **false**.
- Treating the vertex x_3 analogously to x_2 yields that x_3 is colored by **false**, too.

Now for reasons of symmetry, a case analysis for "cl is colored by A_2" and "cl is colored by A_3" gives Table 1.

Table 1

cl	x_1	x_2	x_3
color A_1 implies	**true**	**false**	**false**
color A_2 implies	**false**	**true**	**false**
color A_3 implies	**false**	**false**	**true**

This shows, as a matter of fact, that if we find a valid T-coloring for G, this implies that each clause contains exactly one true literal. Hence, a T-coloring leads to a solution of P.

It remains to show that, reversely, a valid truth assignment for the problem P leads to a valid T-coloring of G. But that is simple: A true literal gets color **true**, a false one gets color **false**. As the s-edge e is colorable by s-edges e_1 and e_2, it is easily checked that this partial coloring can be extended to a coloring of the whole graph G. Thus our proof is complete and we get the desired result. (Did we really want trees to be NP-complete?)

Theorem 5.3. *There is a tree T, for which the T-coloring problem is* NP*-complete.*

6. Discussion

To begin with, the $\underline{X}$-classes do *not* contain all polynomial graphs. The graph $H_8=(V_8,E_8)$, where $V_8=\{1,2,3,4\}$ and $E_8=\{(1,2),(1,3),(2,3),(3,4),(4,3)\}$ (see Fig. 8) is a polynomial graph and does not belong to one of the $\underline{X}$-classes (see [3]).

All families of graphs, for which H-colorability is known to be polynomial time decidable, fit in our concept: The paths $\vec{P}_n$ and transitive tournaments $\vec{T}_n$ are $\underline{X}$-graphs, the cycles $\vec{C}_n$ are $\vec{C}_n$-extended $\underline{X}$-graphs and the semicomplete digraphs with exactly one cycle can be obtained from $\vec{C}_2$ or $\vec{C}_3$ by graft operations.

More important, semipaths and "doublepaths" $\vec{P}_{n,m}$ could be shown to be polynomial by our methods, and this was not previously known. The result for semipaths is "best possible" in the sense that there are already trees which are NP-complete.

One issue we did not address here is the complexity of deciding whether a graph

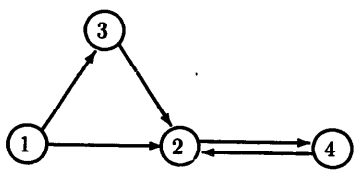

Fig. 8. The polynomial graph H_8 that does not belong to any of the $\underline{X}$-classes.

has the $\underline{X}$-property or the $\vec{C}_k$-extended $\underline{X}$-property. We have seen that an $\underline{X}$-enumeration can be obtained in a straightforward way for semipaths, but no general results are known.

Acknowledgement

The authors thank a referee for a number of suggestions clarifying the presentation. The second author thanks Pavol Hell for discussions on coloring problems.

References

[1] J. Bang-Jensen, P. Hell and G. MacGillivray, The complexity of colorings by semicomplete digraphs, SIAM J. Discrete Math. 1 (1988) 281–298.

[2] M.R. Garey and D.S. Johnson, Computers and Intractability (Freeman, New York, 1979).

[3] W. Gutjahr, Färbung durch gerichtete Graphen, Diplomarbeit, Institutes for Information Processing, IIG, Technical University of Graz (1991).

[4] P. Hell and J. Nešetřil, On the complexity of H-coloring, J. Combin. Theory Ser. B 48 (1990) 92–110.

[5] D.S. Johnson, The NP-completeness column: An ongoing guide, J. Algorithms 3 (1982) 88–99.

[6] H.A. Maurer, A. Salomaa and D. Wood, Colorings and interpretations: A connection between graphs and grammar forms, Discrete Appl. Math. 3 (1981) 119–135.

[7] H.A. Maurer, J.H. Sudborough and E. Welzl, On the complexity of the general coloring problem, Inform. and Control 51 (1981) 123–145.

[8] J. Nešetřil, Representation of graphs by means of products and their complexity, Lecture Notes in Computer Science 118 (Springer, Berlin, 1981) 94–102.